Self-Healing Mechanisms for Kernel System Compromise

Sandy Ring, David Esler, and Eric Cole
Advanced Technology Research Center
The Sytex Group, Inc
Kernel-level “rootkits”

What are rootkits?

*Intruders use rootkit attacks to conceal malicious*

1) processes
2) files
3) network connections

Detection is the hardest aspect
Scope

- Detection ✓
- Forensics ✓
- Recovery ✓
- Adaptation ✗
Out of *current* bounds

Recoverable

- syscall[100]
- syscall[112]
- syscall[116]
- backdoor pid: 34
- hidden files
- read
- sniffer pid: 56
- hidden port traffic

Unrecoverable

- deleted or modified files
- sensitive data
- keystrokes
- attacker
Concept

- **After-the-fact**
- Non-signature dependent
- Comparison of user to kernel behavior
- Loadable kernel module
Implementation

1. System Call Table Analysis
2. Hidden Process Termination
3. Hidden File Removal
4. Attack Traffic Blocking
System Call Table Analysis

- **Problem:** Intruders “patch” addresses in the system call table and redirect applications

- **Solution:** Leverage the Linux kallsyms symbol table to resolve and repair addresses

- **Future Considerations:** Establish call graphs for frequently used functions, perhaps generate signatures?
Hidden Process Termination

- **Problem:** Attackers can hide processes using system call patching, PID 0 renumbering, and task queue removal

- **Solution:** Compare user space to kernel utilization views and terminate discrepancies
Hidden Process Termination

Before:
- hidden process
- parent (if any)
- !(pdeath_signal)
  - memory pages
  - open files
  - file system

After:
- parent (if any)
- hidden process
- pdeath_signal
  - NULL
Hidden File Removal

- **Problem:** This is the hardest problem, the number of ways to hide files is never ending.

- **Solution:** Compare user space listings to kernel file system level records

- **Future Considerations:** Go lower and look for block chains on the drive, but how do you avoid false positives from deleted files?
Attacker Traffic Blocking

- **Problem:** Attackers can hide connections to backdoors which can record and take action on any remediation active you make.

- **Solution:** Immediate disable the communication to suspect processes by monitoring and closing connections via a `dev_add_pack` handler.
Results

- Passed functionality testing
- Minor system degradation (left)
- Negligible network latency (right)
Conclusion

After-the-fact is good, but it should be combined with the adaptability to prevent future attacks all together.
Questions and Suggestions for Improvement?